



ÉCOLE POLYTECHNIQUE DE LOUVAIN

[LSINF 1225]
CONCEPTION ORIENTÉE OBJET
ET GESTION DE DONNÉES

GROUPE F — ANNÉE 2014–2015

TRAVAIL 2

La Conception Orientée Objet

Auteurs :

Mathieu DELANDMETER	6240–13–00
Nathan GILLAIN	7879–12–00
Maxime HANOT	6591–13–00
Alexandre JADIN	4844–13–00
Thomas MARISSAL	8217–13–00
Edouard VANGANGEL	2243–09–00

Professeur :

Kim MENS

Tuteur :

Benoît BAUFAYS

2 avril 2015

Introduction

Dans le cadre du cours de conception orientée objet et gestion de données, dispensé par le Professeur Kim Mens, il nous a été demandé d'implémenter une application *Android* de type "*Gestion de bar*".

Après avoir élaborer une base de donnée en première partie de ce projet, nous nous sommes intéressé à la conception orientée objet.

Ce rapport à pour but de lister et de décrire les différents documents que nous avons créé ainsi que d'expliquer nos choix de conception.

Nous avons pour commencer créé des histoires d'utilisateurs et des cartes CRC, outils qui nous ont permis de créé un diagramme de classe UML et quelques diagrammes de séquences UML. Ces deux derniers documents seront essentiels à l'implémentation du notre application.

Description des documents rendus

Cette section a pour but d'énumérer les différents documents que nous avons produit ainsi que d'en donner une courte description.

Histoires d'utilisateur

Nous avons rédigé toute une liste d'histoire d'utilisateur. Nous les avons réparties en huit catégories :

- Gestion des utilisateurs
- Sélectionner ses préférences
- Afficher l'historique
- Afficher la commande
- Afficher la liste des boissons
- Gestion des stocks
- Gestion de la carte
- Gestion des factures

Cartes CRC

Nous avons réalisé huit cartes CRC. Celles-ci correspondent chacune à une des classes suivantes :

Bien que celles-ci n'aie pas été mises à jour et donc que ce sont des brouillons, elles nous ont permis de bien comprendre comment nous allons organiser notre code. Nous les avons dès lors utilisées afin de créer notre diagramme de classe UML.

Diagramme de classe UML

A partir des cartes CRC, nous avons créé un diagramme de classe UML. Celui-ci modélise toutes les classes que nous implémenterons dans notre application Android. Il reprend les variables et fonctions de chaque classe ainsi que les liens entre ces dernières.

Diagrammes de séquence UML

Nous avons réalisé quatre diagrammes de séquences UML. Ces diagrammes correspondent aux quatre histoires d'utilisateur suivantes :

- En tant que client, afin de consulter la carte, je dois créer un compte protégé par un mot de passe.
- En tant que client, je dois pouvoir consulter la carte des boissons afin de voir la liste des boissons disponibles en tenant compte de mes préférences.
- En tant que serveur, je veux pouvoir gérer les stocks afin de savoir combien de boissons sont encore en stock et donc savoir lesquelles doivent être commandées chez le fournisseur.
- En tant que serveur, je dois pouvoir gérer la facturation afin de faire payer le client.

Choix de conceptions

Cartes CRC

Concernant les cartes CRC, nous avons dans un premier temps créé une carte "*Addition*". Cependant lors de la réalisation du diagramme de classe UML, nous nous sommes rendus compte que celle-ci n'avait pas d'intérêt. En effet, la classe "*Commande*" était suffisante pour gérer tous les cas.

A l'inverse, nous nous sommes rendus compte au même moment que pour pouvoir gérer notre extension "Préférence", nous avons besoin d'une classe "*Historique*". Cette dernière aura une fonction qui s'adapte à l'argument que l'on lui donne. En effet, si elle

est appelée avec un argument de type "Client", l'historique des boissons qui ont été bues par ce client sera affiché. Inversement, si la fonction est appelée avec un argument de type "Boisson", elle affichera l'historique des clients qui ont bu cette boisson.

Utilisation de la base de données

Concernant l'utilisation de la base de données, nous avons décidé de ne pas charger l'ensemble de la base de donnée en instance Java. Nous nous connectons donc à celle lorsque nous avons besoin d'un de ses éléments que nous transformons en instance d'objet Java. Quand nous n'avons plus besoin de celui-ci nous le supprimons.